1996

An Examination of Virtual Reality Modeling Language and its Implications for the Future of the World Wide Web

Mitch L. Hamelau
College of Saint Benedict/Saint John's University

An Examination of Virtual Reality Modeling Language and its Implications for the Future of the World Wide Web

A Thesis

The Honors Program

College of St. Benedict/St. John's University

In Partial Fulfillment

of the Requirements for the Distinction "All College Honors"

and the Degree Bachelor of Arts

In the Department of Computer Science

by

Mitch Hamelau

May, 1996

PROJECT TITLE: An Examination of Virtual Reality Modeling Language and its Implications for the Future of the World Wide Web

Approved By:

J. Andrew Holey

Assistant Professor of Computer Science

Assistant Professor of Computer Science

Assistant Professor of Art

Chair, Department of Computer Science

Director, Honors Thesis Program

Director, Honors Program

Introduction

In the last three years, the computing world has been witness to the phenomenal growth of a new medium of communication. That new medium is the World Wide Web.

The World Wide Web represents a paradigm shift in how people are able to communicate. With the advent of the World Wide Web, people all over the world became able to share thoughts and ideas freely. With the rise of secure transmissions, commerce may also become a common part of how the World Wide Web is used. However, all of this communication is being done in a strictly two-dimensional way; a Web page is flat. At least, Web pages used to be strictly two-dimensional, but a new development tool for the World Wide Web has the potential to completely alter not only how people experience the World Wide Web, but what it is possible for people to communicate on the World Wide Web. That new tool is the Virtual Reality Modeling Language, or VRML. Before launching into a discussion of what VRML is and what it's capable of, let's look at the development of the Internet.

A Brief History of the Internet

The Internet, originally known as the ARPANET (the Advance Research Project Agency NETwork), first came into existence in 1969. The first node was at the University of California at Los Angeles (UCLA). The next three nodes were quickly added. They were at the Stanford Research Institute (SRI), the University of California at Santa Barbara (UCSB), and the University of Utah. In two years, the number of nodes had grown to fifteen, and in 1973 the first two international connections were made to the ARPANET. These connections were the University College of London and the Royal Radar Establishment, in England and Norway, respectively. In 1982, the Transmission Control Protocol (TCP) and the Internet Protocol (IP) were developed as a protocol suite for ARPANET. This suite is commonly referred to as TCP/IP. This development led to the first definitions of an 'internet': "a connected set of networks, specifically those using TCP/IP." The 'Internet' was then defined as connected TCP/IP internets. In 1983, the first 'name server' was developed at the University of Wisconsin. This meant that knowing an exact path to a server was no longer necessary to connect to other systems on the Internet. However, the Internet was still a cryptic, text-based tool that was not very user-friendly. It remained text-based for nearly eight more years and then the World Wide Web was born (Zakon).

In 1991, the World Wide Web, which was developed by Tim Berners-Lee, was released from CERN, the European Laboratory for Particle Physics. This was the first step to providing a more user-friendly way to navigate the Internet. The next big step came in 1993 with the release of Mosaic from NCSA, the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign. Mosaic was the first World Wide Web browser. What it brought to the Web was a method of accessing different sites by using a graphical point-and-click interface. For the first time, navigating the Web was something that could be done by ordinary people with no knowledge of what an URL even was. Beginning in 1993 the already rapid growth of the Internet reached astronomical proportions. The number of hosts on the Internet went from just over one million in 1993 to over nine million by January of 1996 (Zakon). The number of people who have access to the Internet is in the tens of millions world wide. These millions and millions of people are using the Internet in hundreds of different ways.

People are using the Internet as a simple means of sending messages to each other, but they are also using it for much, much more. People are establishing their own personal home pages as a way of sharing information about themselves with people around the world; colleges, universities, libraries, and the Government are putting information on-line for people to access; businesses are keeping their customers up-to-date on their latest product offerings and on the general happenings of the company. The introduction of Java (a secure network programming language) by Sun Microsystems, Inc. in the summer of 1995 brought a new level of possibilities to Web pages: sounds, animations, and interaction. Using Java it is now possible to create Web pages that play sounds automatically when they're loaded, or play animations or react to user actions on the page in some other manner. Java has brought the Web to life. However, we are about to witness an even more profound revolution on the World Wide Web. A seed that was originally planted in 1993 at the First International Conference on the World Wide Web is about to reach full bloom in the form of the widespread use and impending enhancements to the Web language known as Virtual Reality Modeling Language. Let's now look at what exactly Virtual Reality Modeling Language is and see what can be done with it.

What Is VRML?

What exactly is VRML? Here is a formal definition: "VRML is a generic text-based language that describes how to construct 3-D images on the fly" (Hassinger and Erwin 3). Here is a definition that I have come up with after working with VRML: VRML is a language that allows users to browse the Internet in a totally new and exciting way. It allows real-time interaction with 3-D objects and environments over the World Wide Web. It is one of the most exciting things that has happened to the World Wide Web since its creation and it could change how users interact with the Web nearly as much as the creation of the Web changed how users could interact with the Internet.

The ideas that led to VRML were first implemented by Mark Pesce and Tony Parisi in late 1993. It was in 1993 that these two men developed a three-dimensional interface for use with the World Wide Web. They incorporated many of the lessons they had learned from several years of research in the areas of networking and virtual reality (Pesce).

Mark Pesce showed this new interface to Tim Berners-Lee, the developer of the World Wide Web. Berners-Lee was impressed enough that he invited Pesce to present a paper at the First International Conference on the World Wide Web, which was held in Geneva. At the conference, it was agreed that there should be some type of common language to specify 3-D scenes and World Wide Web hyperlinks; sort of a 3-D equivalent to HTML. It was also at this conference that the term Virtual Reality

Modeling Language was first coined. Development of a VRML specification began immediately after the conference had ended (Pesce).

One of the first steps in the development of a VRML specification was to set up an electronic mailing list to provide an opportunity for any and all interested parties to voice their concerns and ideas about what should be entailed in the VRML specification. In less than a week after the mailing list was established, there were over 1000 members. There were three main criteria that were deemed necessary: platform independence; extensibility; and the ability to work using low-bandwidth connections, such as 14.4 kBps modems. Once a set of requirements was agreed upon, the search was on to find technologies that could be used to begin creating this new tool (Pesce).

After examining many alternatives, the Open Inventor ASCII File Format, originally developed by Silicon Graphics, Inc., was selected as the best format to meet all of the requirements that were laid out for VRML. The Inventor File Format provides complete descriptions of 3-D scenes including polygonally rendered objects, lighting, materials, and realism effects. It provides everything that is needed by professionals to create high-quality work, and, since it was an existing format, there was already a large number of existing tools and a wide installed base (Pesce).

Once the decision had been made to use the Inventor File Format as the basis for the VRML specification, Gavin Bell, of Silicon Graphics, began adapting it specifically for VRML. During this process, Bell made use of input from the members of the VRML mailing list (Pesce). The VRML 1.0 specification was completed in the fall of 1994 and was officially announced in October of that year by Mark Pesce, Tony Parisi, and Gavin Bell (Carey).

The first VRML specification provided the groundwork to allow VRML to be used to construct countless objects, scenes and worlds that could be viewed on the World Wide Web using a VRML-enabled browser. (The VRML browser works in conjunction with an HTML browser, such as Netscape Navigator, which continues to handle the 2-D pages.) VRML can be used to render nearly any type of static scene that can be imagined. The VRML 1.0 specification provides an assortment of pre-defined shapes, such as cubes, cones, spheres, and cylinders. These shapes can all have transformations applied to them to shear, scale, or alter their original size, shape, or position. It is also possible to define new shapes by specifying coordinates within a world and connecting them. Each object in a VRML world has a material property which can be set to a solid color or have an image assigned to it. It is also possible to define how an object's surface will interact with light and shadow. Each object in VRML can also be a link to another VRML world or to an HTML page. The VRML 1.0 specification also provides various lighting capabilities. There are three different types of lights: directional, spot, and omni-directional. All of these lights have parameters that can be used to control their color and intensity. The directional and spot lights have parameters for specifying which axes they point along. The omni-directional and spot lights can be positioned anywhere in the world, and the spot light also has controls to adjust the cone angle of its light. This combination of pre-defined and user-defined shapes, the ability to control object material, the ability to link to other worlds or pages, and the ability to position the needed type and color of lights within a world gives VRML incredible flexibility and power in what it can be used to create. However, despite all of the capabilities it currently has, there are still limitations to what is currently possible in VRML.

The biggest limitation of the VRML 1.0 specification is that it only provides the capability to create static worlds. Other than being able to use an object in a world as a link to another world, there is no user-interaction with the objects in a world. It's true that the user can rotate a world or move around within the world, but the user cannot interact with individual objects in a world. Even more limiting is the fact that objects in a world cannot be given some type of inherent behavior by the world designer. VRML also lacks support for sound in its current specification. So, despite the fact that VRML does allow the creation of complex and realistic-looking worlds, they are all stationary and silent. Fortunately, all of the limitations of the VRML 1.0 specification are being addressed by the VRML 2.0 specification.

Even though the VRML 2.0 specification is still in development, it is already being greatly anticipated by the on-line community. The approach that was taken to begin development of the VRML 2.0 specification was to send out a request for proposals and see what came back. A large number of companies and organizations, including IBM, SGI, Microsoft, and Apple, responded to the call for a new specification and submitted their proposals for the next version of VRML. The proposal that won the most support and will be the basis for VRML 2.0 is the one that SGI was involved in developing: Moving Worlds (Wilcox 36). Moving Worlds will add four main enhancements to VRML 1.0: more realistic static worlds, user interaction, animation, and prototyping (What Moving Worlds Is All About).

With Moving Worlds, you would have ground-and-sky backdrop capabilities. You could add things to a world such as mountains, clouds, and fog. You would also be given the ability to create irregular terrain for the ground instead of being limited to flat planes. You could also add sounds, such as crickets, into a world to give an even more realistic effect (What Moving Worlds Is All About).

Moving Worlds allows true interaction with objects in the world. You can interact directly with objects in a world by clicking them or moving within a certain distance of them. There is also collision detection, so objects that are supposed to be solid in a world, such as walls, will behave like solid objects. There is also a sensor that can be used to keep track of time passage for controlling different objects in a world. This time sensor could form the basis for creating animations, such as a clock's hands moving or birds flying, within a world (What Moving Worlds Is All About).

The prototype abilities that Moving Worlds provides allow the sharing and re-use of user-created objects or scripts. If somebody developed a robot object, for instance, all of the nodes and behaviors that make up the robot could be encapsulated as a new type of node called Robot. This node could then be used in many different worlds and even by different developers (What Moving Worlds Is All About).

It should rapidly become apparent that the next version of VRML will do more than remove the current limitations of VRML, it will make entirely new things possible.

Future Possibilities in VRML

One of the most dramatic changes that VRML could bring about is the manner in which the World Wide Web is navigated. The Web is very much a non-linear information space at heart. There is no uniform path that needs to be followed to get from one site or subject to another; that's part of what makes the Web such a personal thing for the people who use it: on the Web, more than in any other communication medium, the user has nearly total control over where they go and how they get there. Up to this point, the only real option for navigating the Web in a user-friendly manner has been through HTML pages, and they have done a very admirable job so far. However, there has always been a dimensionality limit to them: the Web is three-dimensional, but HTML pages are only two-dimensional. The hyper-linking capabilities of HTML have done a good job of linking together different sites on the Web, but these links are limited to being laid out in a linear fashion. The continuing development, and spreading use, of VRML has the potential to change all of that.

By using a three-dimensional interface for browsing the Web, it would become even more intuitive for users to work their way through the multitudes of information that comprise the Web. It would be possible for sites to be laid out in ways that could more effectively communicate to a visitor what the various elements of a site were, and also more clearly show the relationship between elements of the site and show the relationship between the site and the Web as a whole. For example, if you reached a site that was somebody's personal homepage (Figure 1), maybe you would encounter a small house with the person's name on a sign in front of it and a Welcome Mat inviting you inside. You could then actually move inside of the house and go to various rooms to see what the person has on their site.

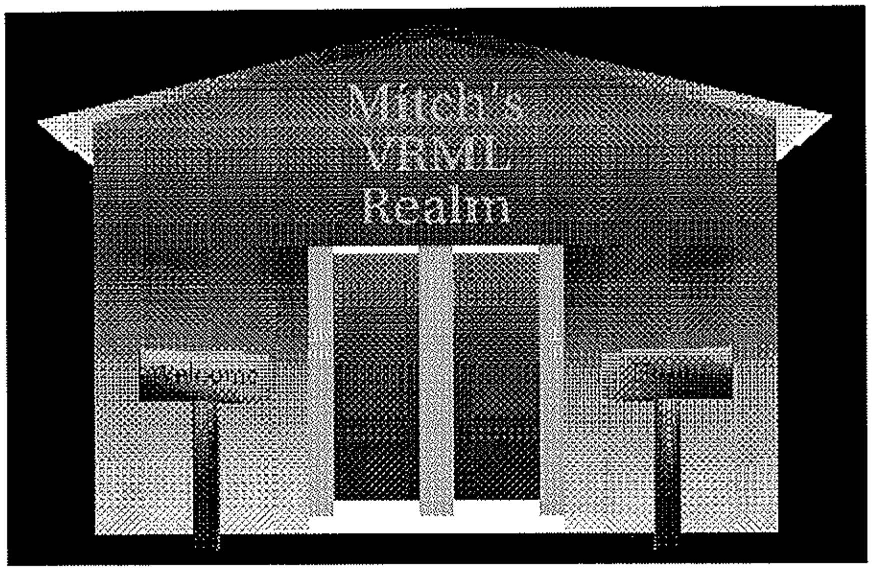

Figure 1: Example VRML homepage

There might be a Family Room where you could find pictures of the person's family sitting on a mantel, with various tidbits of information about the person and the person's family members scattered throughout the room. Or maybe you walk into the Living Room to discover which movies, songs, and television shows are among the person's favorites. There could even be a television and a stereo in the room on which you could hear small samples of the person's favorite songs or watch clips from their favorite movies. The possibilities are innumerable. This same type of approach could be used by corporate sites as well. Instead of walking into a house, you could walk into the Corporate Headquarters and take an elevator to the floor that had the information you were looking for. This type of navigation on the Web would be far more exciting and intuitive than the simple HTML pages of today. Not only would this be a more intuitive approach for users, but it would open up a whole new realm of possibilities with regards to what corporate sites could offer on the Web.

If you were to arrive at Joe's Furniture Emporium (Figure 2), for example, you could do far more than just get some information about Joe's fine selection of products

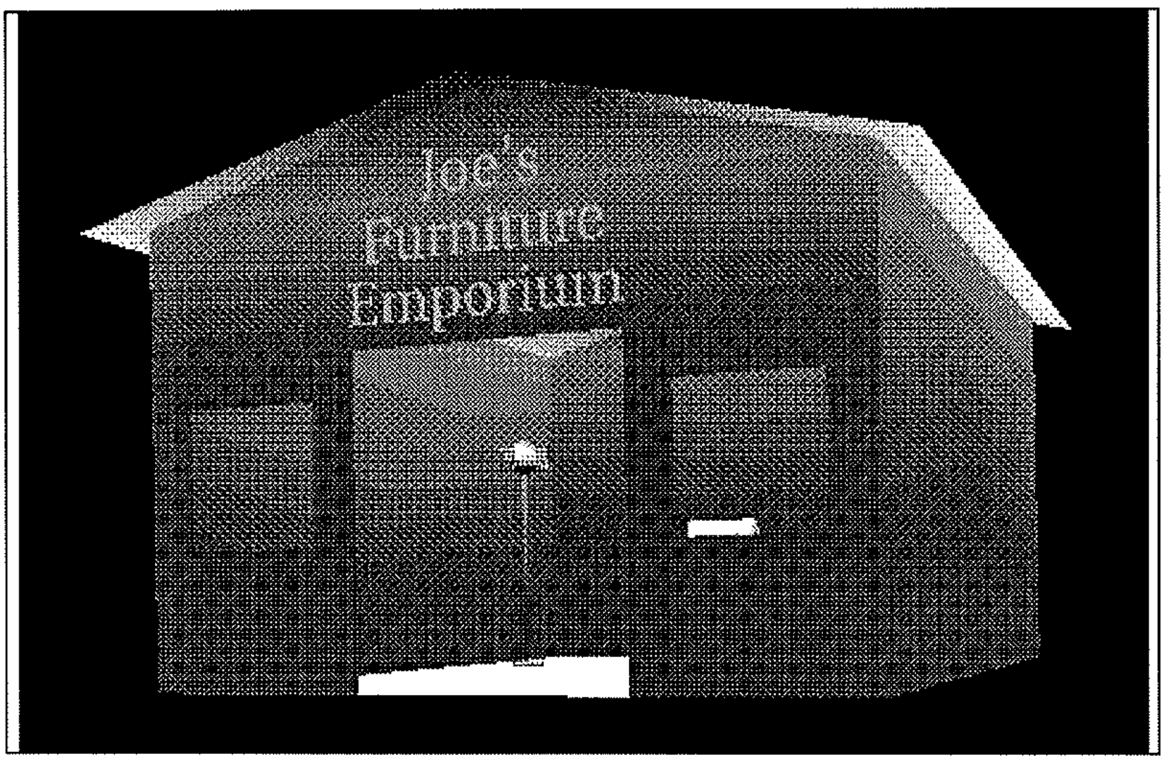

Figure 2: Joe's Furniture Emporium VRML Site

and look at some GIF images of the products. You could actually walk in the front door of Joe's Furniture Emporium and be greeted by a showroom that contained all of the various pieces of furniture that Joe's had to offer. You could then walk around the showroom floor and look at all of the various pieces. When you found something you liked, you could click on it and a close-up of just that item could be loaded. Then, if you clicked on the close-up model, a screen of all relevant information, such as materials and prices, could open in an HTML browser. If you were really interested in the piece of furniture, you could use the store's mannequin to try it out for you. All you would have to do would be to feed in your measurements (height, waist size, etc.) and the mannequin could take on your proportions and sit in the furniture for you. You would actually have the ability to see how your own body (at least a close approximation, anyway) looked in the furniture. It would even be possible in the future to have an actual model of yourself that you could use for trying out furniture and other items, such as clothing. This far surpasses anything that is even possible right now. It would be a way of providing far more information to a visitor in a far more natural and intuitive way than HTML pages could ever achieve.

As the example of Joe's Furniture Emporium demonstrates, the whole realm of on-line shopping could be revolutionized by VRML. Any type of company that sells physical items could show them off completely in their VRML world. The new designs of on-line malls would far surpass any of the current implementations. It would be possible to have an actual VRML mall world that you could walk around in and see the various storefronts. When you found a store you were interested in, you could walk inside it and view whatever they had to offer you.

The possibilities that VRML can bring to the Web go far beyond strictly retail-oriented businesses. Imagine being able to contact an architectural firm anyplace in the world, for example, and having them design a house for you that you could walk through and see with furnishings of your choosing. It's true that such things are already being done on computers, but you have to use the architect's computer to do it, which means you're limited by location. Using VRML, the architect could be located in any corner of the world and it wouldn't make any difference.

Another area where VRML holds potential is tourism. Cities could set up virtual representations of themselves, their key attractions, or both, that people could visit on the Web. It would help take a lot of the guess work out of travelling for tourists because they would have the chance to see their destinations before they went there. These representations could be made very accurate. It would be possible to have moving vehicles, or animals, in the world, for instance. If there was a stream in the world, it could actually be moving and have ripples and waves on its surface, along with the sound of gurgling water which would get louder as a visitor got closer to the stream. Scattered around in the world could be representations of unique stores or attractions that the visitor could go into or interact with, respectively. There is already an example of this type of application. It is a little tour of Maui. While lacking in detail and interaction (due to the limitations of VRML 1.0) it provides a good starting point to help envision what could be possible in the near future.

It has been established that VRML will offer on-line developers and consumers many advantages in the new applications that can be developed, but what about existing applications? VRML will provide many opportunities for improving on them also.

A specific example of how VRML could be used to implement an existing application in the realm of 3-D is the SnowBall application.

SnowBall: A Concrete Example

SnowBall is the development name of a new multimedia presentation package. This application allows the integration of computer-generated images, 35mm slides, audio, lighting, and other devices into a single package. It provides a presenter with much more control over the presentation environment than a computer-only package does. SnowBall will make it possible for a presenter to control all aspects of a presentation using a single package.

Here is an outline of the steps involved in creating a presentation using SnowBall:

1) Establish a 'logical' representation of the presentation environment.
2) Group similar devices (Projectors, Lights, etc.).
3) Place commands for the various devices on the timeline.
4) Physically hook presentation computer up to all of the necessary devices and test the presentation.
5) Make any necessary adjustments to functions on the timeline, while still in the presentation room.

As stated in the outline above, the approach that SnowBall uses for setting up a presentation is to 'logically' group devices according to their position in the actual presentation environment. For example, if two projectors' displays were going to overlap each other, they would be grouped into a single 'Visual Area'. The same type of approach is used with the lights and other devices in the presentation environment. This approach provides the presenter with a more complete representation of the environment than simply having a wide assortment of individual devices to control.

The environment approach SnowBall uses also allows different presentations to be developed with the appropriate set-up in mind. For example, if a lot of presentations are given in a boardroom, then a Boardroom environment can be created with the appropriate number of logical areas and the correct devices assigned to each area. Then, anytime somebody needs to prepare a presentation for the boardroom, they can use the Boardroom environment to ensure that their presentation will take full advantage of the capabilities of the actual boardroom.

Once the individual devices have been grouped into logical areas, the different functions they are to perform can be determined and laid out on a timeline where they can be re-positioned for fine-tuning to match certain images to certain sections of a soundtrack, for example.

After all of the functions have been laid out and fine-tuned, the presentation can be played back and all aspects of the environment will be controlled exactly as they were laid out on the timeline. The actual control of the devices is achieved using a standard RS-232 connection between the controlling computer, the projectors, and the appropriate adapters for the rest of the devices that are needed for a given presentation.

SnowBall is being developed on Intel-based PCs running Windows NT 3.51.

The target platforms for the completed project are Intel-based PCs with at least a 486 processor and either the Windows NT 3.x or Windows 95 operating system with a minimum of 16M of RAM. The target customers for SnowBall would include anybody who needs to give effective multimedia presentations, but the main focus is on mid- to large-sized companies and institutions.

A product such as SnowBall has tremendous potential for being moved on-line using a tool like VRML. The advantages of such an implementation are many.

SnowBall is trying to provide the presenter with as complete a picture of the presentation environment as possible. However, the current representation is limited to a 'logical' 2-D representation of the layout of the presentation environment. While still providing advantages over the individual-device approach, this technique is still not ideal. Ideally, the presenter should be able to see a representation of the actual environment the presentation is being designed for. VRML is exactly what's needed to accomplish this.

Here is an outline of the steps needed to create a presentation using a VRML 2.0 implementation of SnowBall followed by a more detailed description of how the presentation would be built in this new version of SnowBall:

1) Create an accurate 3-D model of the real-world presentation environment.
2) Use controls on model devices that correspond to the actual devices in the presentation room to place functions on the timeline.

3) Run a full preview of the presentation, from any local networked computer, that will show what each device does.

4) Make any necessary revisions.

5) Give your presentation with full knowledge of how everything is going to work in the presentation room.

Using VRML 2.0, it would be possible to create scale-model representations of any conceivable environment that a presentation was being designed for. There could be pre-defined environment shells of some standard room layouts, such as a Boardroom (Figures 3 and 4), that could be used as templates for creating the actual environment that the presentation was being designed for.

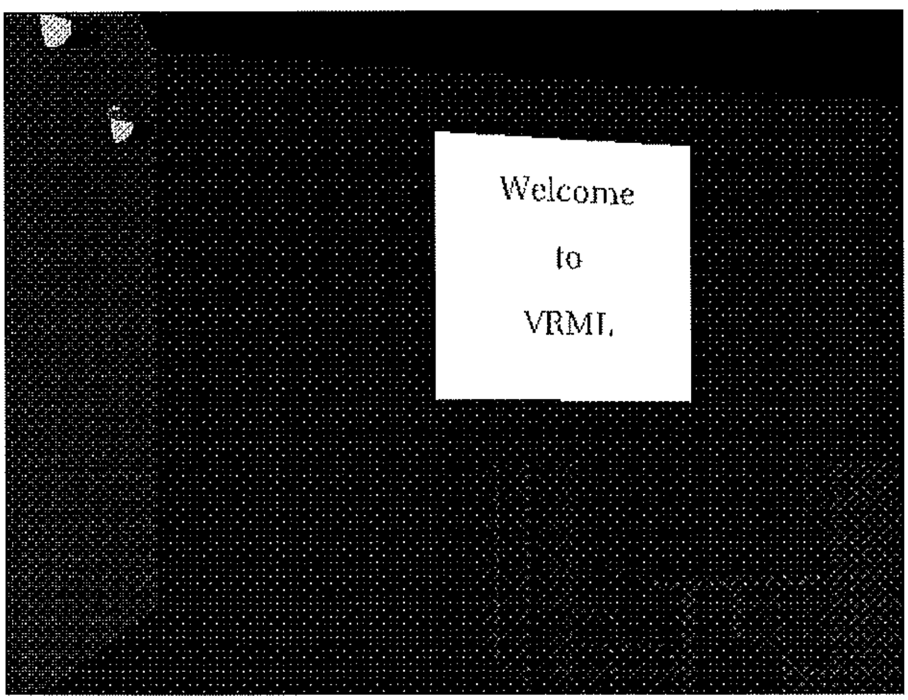

Figure 3: Looking into 3-D Boardroom Environment

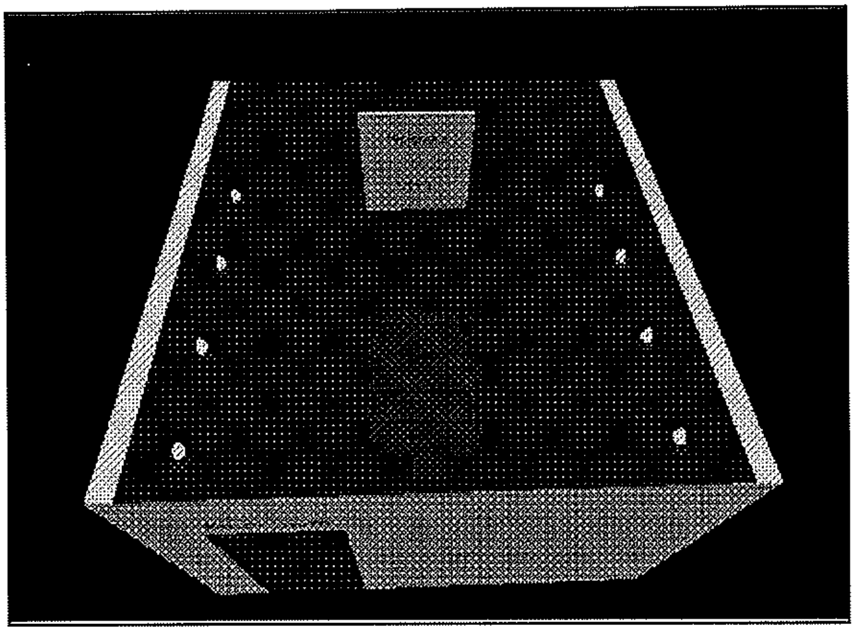

Figure 4: Top view of 3-D Boardroom Environment

Nearly all aspects of the environment could be adjusted, such as: wall sizes; ceiling height; the number of windows and their locations; door locations; the number of seats and their locations; the number of projector screens and their locations; the number, position, and direction of projectors; the number, location, color, direction, and style of lights; etc. Using a combination of VRML 2.0's ability to allow interaction with objects in a world and Drag and Drop techniques, the presenter could be given an Environment Design world where they could construct the exact environment they needed to design a presentation for. All of the elements in the environment could have very realistic representations, too. For instance, under the lighting tools, there could be a Table Lamp, a Ceiling Light, a Spot Light, etc. Each of these tools would look like the type of light it was representing in the real-world presentation environment. These tools could then be placed wherever they were needed in the world to depict the actual environment of the presentation room. The same techniques could be used to position windows, doors, chairs, tables, projector screens, projectors, etc. Any element in the actual environment could be placed in the model environment at the correct location. And each of these elements would have the appearance of their real-world counterparts.

Already it becomes apparent that this type of approach for designing a presentation is far more intuitive than any other method used by any existing presentation system. There is no longer any doubt about what device is being controlled at any given point or how everything will work once it is played in the actual presentation environment. The presentation designers will know ahead of time exactly what devices they have to work with and will actually be able to see their relationship to each other and the environment they are contained in. The next question becomes, How would the presenters begin laying out the presentation once they have the devices all correctly arranged? You press the appropriate button on the appropriate device.

Since the model environment is an accurate representation of the actual presentation environment, and each of the devices contained in the model environment is an accurate representation of the actual device, the most natural, intuitive way for a presenter to begin laying out a presentation would be to make use of the controls that are on each of the various devices. For instance, clicking on the model projectors could open control panels that had the same controls on them as the actual projectors would. They could have buttons for advancing to the next slide, backing up to a previous slide, adjusting the lamp level, etc. So to make Projector1 advance to the next slide, it would be a simple matter of clicking on Projector1 and then pressing the Advance button on the Projector1 control panel. This same straight-forward approach could be used with all of the various devices contained in the presentation environment. There would be no uncertainty left in the presenter's mind about what device was being told to do something because the presenter would actually 'see' the device that was being controlled.

Once a presentation had been laid out, it would be possible to run a complete preview of the presentation. If the images that were being used in the presentation were available in a digital format, they could actually be 'projected' in the model environment. This means that an entire presentation could be built for a particular presentation environment and fully previewed for that same environment without any hardware other than the computer being used to design the presentation and without even having to be in the actual environment itself.

There are still some design issues that would have to be dealt with in implementing SnowBall in VRML. For instance, the issue of the toolbox. It would be possible to place the toolbox in the VRML world itself, but would there be an advantage to doing this or would it be more convenient to place the toolbox on a companion HTML page or in a separate VRML world? Placing the toolbox in the main VRML window would provide the advantage of close proximity between tool and destination, but it would also use up more viewing volume in an already constrained space. Placing the toolbox in a companion VRML world would solve the problem of using up space in the main window, but would it provide any real advantage over placing the toolbox on an HTML page? An HTML page would restrict the tools to being 2-D objects until they were placed in the VRML world, but would there be a need to have 3-D representations of the tools in the toolbox in the first place? Similar types of issues arise when dealing with representing a timeline for the project. At this point, I think that a companion HTML window may be the best solution because it also has the advantage of being able to more easily convey a larger amount of textual information to guide the presentation designer. However, deciding on a preferred approach will require actual implementation of the various methods to see what works best once VRML 2.0 is fully developed.

Design details aside, this type of approach to a presentation package is quite unique and completely intuitive for the user, but uniqueness and an intuitive interface would not be the only advantages that a VRML implementation of a software application would offer.

In its current implementation, SnowBall, and all software applications in general, has to be designed to work for a specific operating system. This is very limiting and costly. It means that only people who have computers that are running the appropriate operating system are able to use the application. To allow other operating systems to run the application, it would have to be ported to each one individually. This itself can be a major undertaking, but it gets worse. As the different operating systems are changed and updated, the application may need to be rewritten just to run on the new operating system. Both of these problems can be greatly minimized, or even eliminated, by using VRML as the implementation platform.

VRML was designed to be platform independent, just like HTML. As long as the user has the necessary browser for their computer, they can view any VRML world regardless of the type of machine they are using or the type of machine the world was developed on. A VRML application would achieve device independence, which means that anybody with a computer, a modem (or other network connection), and a browser is now a potential customer. The application never needs to be ported for a different operating system. The on-line approach would also simplify the updating of an application. Updating an application would become just that: updating the application to take advantage of any new capabilities of VRML, not completely re-working an application just to make it continue to function correctly. It wouldn't even be necessary to alter the application to keep it running, it would just be something that was possible. Another advantage of using an on-line approach to software application development would be the ease of distribution.

In a traditional approach to software design and development, the finished application has to be shipped to the consumer on floppy disks or CD-Roms. The same thing applies to updated versions of these applications: the consumer must purchase and be shipped an entire new version of the application to take advantage of any new capabilities. This whole process can be completely removed by the on-line approach.

If an application, such as SnowBall, were developed on-line, no shipment of the product would be necessary. All it would require for a consumer to use the application would be to go to the appropriate location on the World Wide Web. This also means that any revisions to, or updates of, the application would be completely transparent to the user. Any time the consumer went to the application's site on the Web, they would always be accessing the most up-to-date version of the application without having to do anything out of the ordinary. This approach has the potential to revolutionize both the pricing and distribution of software applications.

SnowBall is only one example of how an existing application could be migrated to the World Wide Web using VRML to the advantage of both the software developer and the consumer. There are many other applications that could readily take advantage of the 3-D capabilities of VRML and the World Wide Web. Any type of application that involves some type of 3-D visualization or that would be improved by being able to make use of 3-D visualization is a prime candidate for an on-line implementation in VRML. There are many software applications that fall into those categories. Picture a File Management application where you could actually 'walk' to the filing cabinet to retrieve a particular document. Imagine being able to always play the most recent version of Doom, or a version of Myst where new puzzles and worlds were constantly being added. The potential for software developers is limited only by their imaginations.

Conclusion

Many of the ideas discussed here are not yet completely possible, but they are all very realistic when framed in the context of the developments that are currently underway in many areas of the computing world. The next few years are going to see changes in how people and computers interact that will make the changes of the last few years seem minuscule by comparison. The Internet, and especially the World Wide Web, will become more and more a part of people's everyday lives. The way business is conducted will continue to evolve and become more focused towards on-line delivery of content to consumers. I believe that VRML, because of its ability to allow users to interact with objects in more intuitive ways than HTML could ever achieve, will be at the forefront of many of these changes. I believe that the combination of VRML and the World Wide Web will provide the next step toward removing the user's shift between interacting with the outside world and interacting with a computer. The continued decrease of this mental shift will enable more and more people around the world to use computers in all aspects of their lives.

Works Cited

Carey, Rikk. "Stumbling into Cyberspace." http://www.sgicom/ion/Spring_95/vrml.ezine.3.95.html Last accessed: 6 May 1996.

Hassinger, Sebastian, and Mike Erwin. Mecklermedia's Official internetWorld 60 Minute Guide to VRML. Foster City: IDG Books Worldwide: 1995.

Pesce, Mark. "History of Virtual Reality Modeling Language." http://webspace.sgi.com/Archive/VRML-history/index.html Last accessed: 6 May 1996.

"VRML 1.0 Specification." http://vrml.wired.com/vrml.tech/vrml10-3.html Last accessed: 6 May 1996.

"VRML 2.0 Specification." http://webspace.sgi.com/moving-worlds/spec/spec.main.html Last accessed: 6 May 1996.

"What Moving Worlds Is All About."

http://webspace.sgi.com/moving-worlds/over3/summary.html

Last accessed: 6 May 1996.

Wilcox, Sue. "VRML 2.0 Takes Flight." NewMedia 1 April 1996: 36.

Zakon, Robert H. "Hobbes' Internet Timeline v2.4a."

http://info.isoc.org/guest/zakon/Internet/History/HIT.html

Last accessed: 6 May 1996.